Q1 Commands
5 Points

List the commands used in the game to reach the first ciphertext.

climb, read, enter, read

Q2 Cryptosystem
5 Points

What cryptosystem was used in this level?

A Simple Substitution Cipher(Monoalphabetic)

Q3 Analysis
25 Points

What tools and observations were used to figure our the cryptosystem? (Explain in less than 100 words)

On the 1st screen, there was a trail saying "seemingly leading to the top of the hill". So we tried the command "climb" as to climb the hill.
The 2nd screen told us that there was some message written on the blocks, so the command "read" made sense for reading the message.
The 3rd screen directly told us to write "enter",
Again it mentioned that something was written, so we used "read" again.
Initially looking at the cipher, it looked like some random text having some patterns. We saw that there were some repeated sequences like "Fic" and "ok". This gave us an idea that we should try shift cipher and substitution cipher. We tried shifting with 1-25, but the message still looked random for each case.

Digram and Trigram analysis done on cipher text helped us to find common patterns that were present in text for example digrams like(fi->th,ic->he,cm->er,ck->es,oq->in,fc->te,kf->st,pi->ch,mc->re,io->hi,ok->is,hd->am,dn->mb) and trigrams like(fic->the,hdn->amb,fio->thi,gef->oft,icp->hec,hsc->ave,cfi->eth,mco->rei,qfc->nte,fcm->ter,omk->irs,mkf->rst,cmg->ero) after doing such analysis it made cipher a little bit more clearer but still cipher was not comprehensible hence we dig further deeper to gather more insights.
Then we tried doing frequency analysis for substitution cipher along with known plaintexts for solving it. The code attached named "freqAnal.py" does the frequency analysis of all the letters present in the ciphertext. Frequency analysis of cipher text is:
[('C', 13.95), ('F', 10.85), ('K', 10.47), ('O', 9.69), ('I', 8.53), ('G', 5.43), ('H', 5.04), ('M', 5.04), ('Q', 4.65), ('P', 3.49), ('D', 2.71), ('N', 2.71), ('V', 2.71), ('E', 2.33), ('Y', 2.33), ('A', 1.94), ('U', 1.94), ('J', 1.55), ('L', 1.55), ('R', 1.16), ('X', 1.16), ('S', 0.78), ('B', 0.0), ('T', 0.0), ('W', 0.0), ('Z', 0.0)]
'e' is the most frequent letter in english and in our case, 'c' had the highest frequency. So, c was mapped to e. Similarly, 'f' was mapped to 't' which was 2nd highest occuring letter in the ciphertext.
So we assumed Fic as Tie and as it was the highest occuring, we considered Tie=The. So i mapped to h.
Using known plaintext and the knowledge of digrams and trigrams,we assumed thehe=there, ok=is and we mapped o=>i, k=>s,h=>r.
We observed that the characters [. , !] were not substituted.
Finally, it was written that the digits have been shifted by 2 places. But as the 2 itself in the sentence might be the shifted output, we tried all possible shifts from 1 to 9 for the digits inside the password. Finally, a shift of +4 worked(So, the digits were shifted by 4 backwards while encrypting). The password that we got was iRqy3U5qdgt.

Q4 Mapping

10 Points

What is the plaintext space and ciphertext space?
What is the mapping between the elements of plaintext space and the

elements of ciphertext space? (Explain in less than 100 words)

ciphertext space(b,t,w,z absent)->
{a,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,u,v,x,y,1,2,9}
plaintext space (j,q,x,z absent)->
{a,b,c,d,e,f,g,h,i,k,l,m,n,o,p,r,s,t,u,v,w,y,3,5,6}

Mapping from cipher text to plain text:
a->g,c->e,d->m,e->f,f->t,g->o,h->a,i->h,j->p,k->s,l->w,m->r,n->b,o->i,p->c,q->n,r->y,s->v,u->l,v->u,x->q,y->d,9->3,1->5,2->6.

Q5 Password

5 Points

What is the final command used to clear this level?

iRqy3U5qdgt

Q6 Codes

0 Points

Upload any code that you have used to solve this level

Crypto_assign_1.py Download

```
1  letterFreq = {'E': 12.70, 'T': 9.06, 'A': 8.17, 'O': 7.51,
   'I': 6.97, 'N': 6.75, 'S': 6.33, 'H': 6.09, 'R': 5.99, 'D':
   4.25, 'L': 4.03, 'C': 2.78, 'U': 2.76, 'M': 2.41, 'W': 2.36,
   'F': 2.23, 'G': 2.02, 'Y': 1.97, 'P': 1.93, 'B': 1.29, 'V':
   0.98, 'K': 0.77, 'J': 0.15, 'X': 0.15, 'Q': 0.10, 'Z': 0.07}
2
3  letters=list('ABCDEFGHIJKLMNOPQRSTUVWXYZ')
4
5  cipher= """omkf pi hdn cmgef icphsck .H krg vphqkc c,
6
7  fic mco kqgf ioqag eo qfcmckf oq ficpihdn
8
9  cm .Kg dcgeficu hfcm pi hdn cmklo uuncdgmc
10
11 oqfc mc kfoq afihqfiokgq c!Fi cpgy cvkc yeg
12
13 mfio kdck kha cokh kodiuck vn k fofvfo
```

```
14
15  gqpojicmoqli opiyoa of kihsc nccqki oefc
16
17  ynr2 juhpck. Fi c jhkklgm yok oMxr9V1x ya
18
19  flofigvffic xvgfck. Fio kokfice"""
20
21
22  cipher=cipher.upper()
23
24  cipFreq={}
25
26  mapping={}
27
28  total=0
29
30  for let in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ': # Calculate total
    letters
31
32          total+=cipher.count(let)
33
34  for let in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ': # Do Frequency
    Analysis
35
36          cipFreq[let]=round(cipher.count(let)*100/total,2)
37
38
39  letterFreq=list(letterFreq.items())
40
41  cipFreq=list(cipFreq.items())
42
43  cipFreq.sort(key=lambda x: x[1], reverse = True)
44
45
46  for i in range(6): # Create the mapping dictionary
47
48          mapping[cipFreq[i][0]]=letterFreq[i][0].lower()
49
50  print("CIPHERTEXT:  ")
51  print("{}".format(cipher))
52
53  plain=cipher
54
55  mapping['I'] = 'h'
56
57  mapping['K'] = 's'
58
59  mapping['O'] = 'i'
60
```

```
61  mapping['M'] = 'r'
62
63  mapping['E'] = 'f'
64
65  for key in mapping:
66
67          plain=plain.replace(key,mapping[key]) # Replace the
    substituted characters from frequency analysis.
68  print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
69  print("English Frequency: {}\n".format(letterFreq))
70  print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
71  print("Cipher Frequency Analysis: {}\n".format(cipFreq))
72  print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
73  print("Half Plaintext:\n{}".format(plain))
74
75  print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
76  print("[+]Plaintext using digraph/trigraph analysis and manual
    inspection :\n","IRST CH AMB EROFT HECAVES .A SYO UCANSE E,
    THE REI SNOT HINGO FI NTEREST IN THECHAMB ER .SO MEOFTHEL ATER
    CH AMB ERSWI LLBEMORE INTE RE STIN GTHANTHISON E!TH ECOD EUSE
    DFO RTHI SMES SAG EISA SIMPLES UB S TITUTI ONCIPHERINWH ICHDIG
    IT SHAVE BEENSH IFTE DBY2 PLACES. TH E PASSWOR DIS iRqy9U1q dg
    tWITHOUTTHE QUOTES. THI SISTHEF")
77  print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
78
79  #USING FINAL SUBTITUTIONS as
80
81  #[ABCDEFGHIJKLMNOPQRSTUVWXYZ]
82
83  #[GKEMFTOAHPSWRBICNYVJLUXQDZ]
84
85
86  '''Final plaintext that we got after rearrangement of spaces
    is:
87
88  IRST CHAMBER OF THE CAVES. AS YOU CAN SEE, THERE IS NOTHING OF
    INTEREST IN THE CHAMBER. SOME OF THE LATER CHAMBERS WILL BE
    MORE INTERESTING THAN THIS ONE! CODE USED FOR THIS MESSAGE IS
    A SIMPLE SUBSTITUTION CIPHER IN WHICH DIGITS HAVE BEEN SHIFTED
    BY 2 PLACES. THE PASSWORD IS iRqy9U1qdgt WITHOUT THE QUOTES.
    THIS IS THE F
89
90  '''
91
92
93
94  #After Getting the password, try permutations of shifted
    numbers in circular way(mod 10) and attempt the password.
95
```

```
cipher=list("iRqy9U1qdgt")

print("Cipher={}".format(cipher))

for i in range(1,10):

        cip=list(cipher)

        cip[4]=str((int(cip[4])+i)%10)

        cip[6]=str((int(cip[6])+i)%10)

        print("Number Increment Shift
{}".format(i),''.join(cip))

```

Q7 Team Name

0 Points

INSYNC

Assignment 1

GRADED

GROUP

Punit Chaudhari
Piyush Gangle
Aman Mittal
View or edit group

TOTAL POINTS

45 / 50 pts

QUESTION 1

Commands **5** / 5 pts

QUESTION 2

Cryptosystem **5** / 5 pts

QUESTION 3

Analysis **20** / 25 pts

QUESTION 4

Mapping **10** / 10 pts

QUESTION 5

Password **5** / 5 pts

QUESTION 6

Codes **0** / 0 pts

QUESTION 7

Team Name **0** / 0 pts